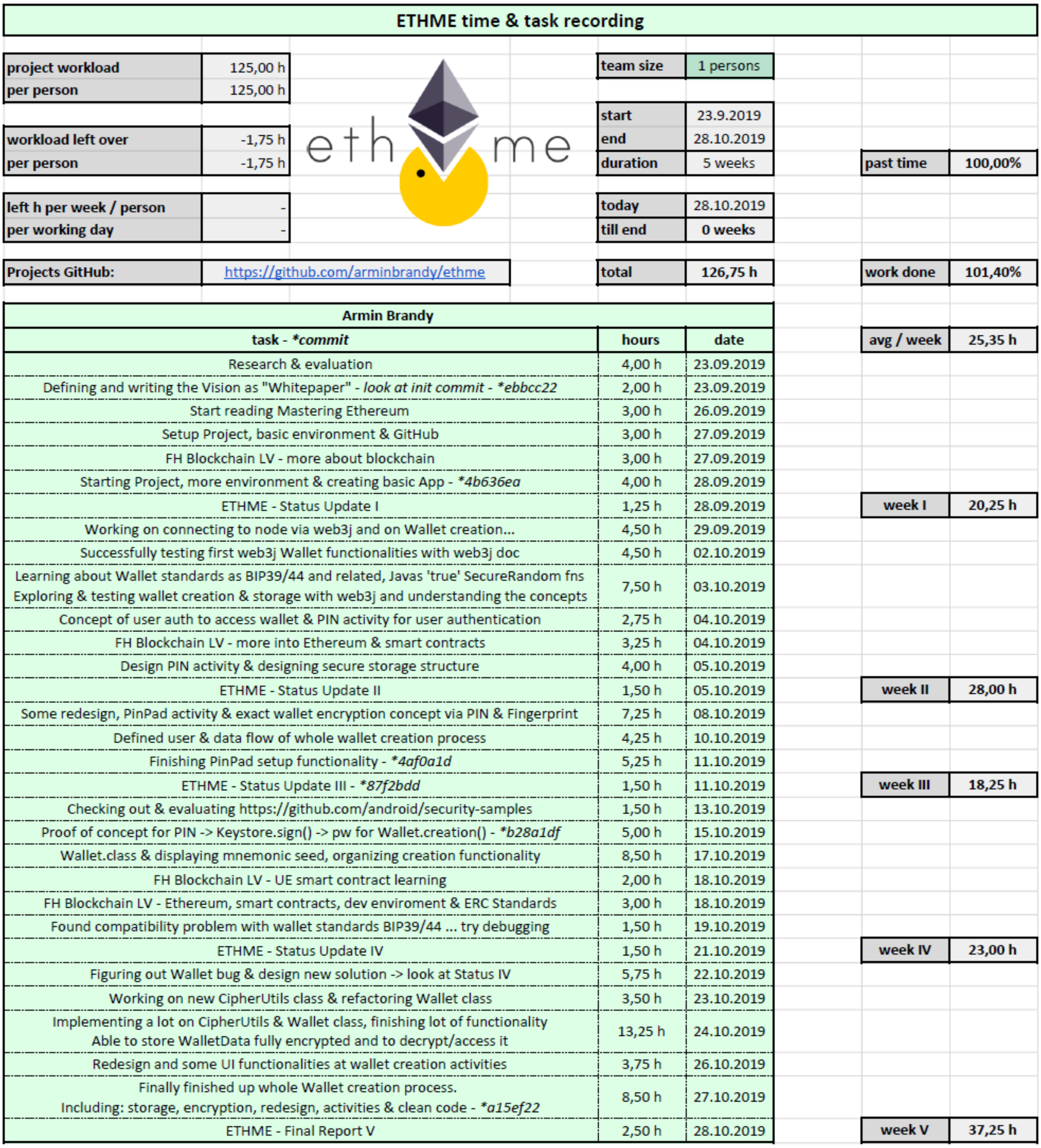
ETHME – Final Report



ETHME time & task recording

project workload	125,00 h	team size	1 persons		
per person	125,00 h				
workload left over	-1,75 h	start	23.9.2019		
per person	-1,75 h	end	28.10.2019		
		duration	5 weeks	past time	100,00%
left h per week / person	-	today	28.10.2019		
per working day	-	till end	0 weeks		
Projects GitHub:	https://github.com/arminbrandy/ethme	total	126,75 h	work done	101,40%

Armin Brandy

task - *commit	hours	date		
			avg / week	25,35 h
Research & evaluation	4,00 h	23.09.2019		
Defining and writing the Vision as "Whitepaper" - look at init commit - *ebbcc22	2,00 h	23.09.2019		
Start reading Mastering Ethereum	3,00 h	26.09.2019		
Setup Project, basic environment & GitHub	3,00 h	27.09.2019		
FH Blockchain LV - more about blockchain	3,00 h	27.09.2019		
Starting Project, more environment & creating basic App - *4b636ea	4,00 h	28.09.2019		
ETHME - Status Update I	1,25 h	28.09.2019	week I	20,25 h
Working on connecting to node via web3j and on Wallet creation...	4,50 h	29.09.2019		
Successfully testing first web3j Wallet functionalities with web3j doc	4,50 h	02.10.2019		
Learning about Wallet standards as BIP39/44 and related, Javas 'true' SecureRandom fns Exploring & testing wallet creation & storage with web3j and understanding the concepts	7,50 h	03.10.2019		
Concept of user auth to access wallet & PIN activity for user authentication	2,75 h	04.10.2019		
FH Blockchain LV - more into Ethereum & smart contracts	3,25 h	04.10.2019		
Design PIN activity & designing secure storage structure	4,00 h	05.10.2019		
ETHME - Status Update II	1,50 h	05.10.2019	week II	28,00 h
Some redesign, PinPad activity & exact wallet encryption concept via PIN & Fingerprint	7,25 h	08.10.2019		
Defined user & data flow of whole wallet creation process	4,25 h	10.10.2019		
Finishing PinPad setup functionality - *4af0a1d	5,25 h	11.10.2019		
ETHME - Status Update III - *87f2bdd	1,50 h	11.10.2019	week III	18,25 h
Checking out & evaluating https://github.com/android/security-samples	1,50 h	13.10.2019		
Proof of concept for PIN -> Keystore.sign() -> pw for Wallet.creation() - *b28a1df	5,00 h	15.10.2019		
Wallet.class & displaying mnemonic seed, organizing creation functionality	8,50 h	17.10.2019		
FH Blockchain LV - UE smart contract learning	2,00 h	18.10.2019		
FH Blockchain LV - Ethereum, smart contracts, dev enviroment & ERC Standards	3,00 h	18.10.2019		
Found compatibility problem with wallet standards BIP39/44 ... try debugging	1,50 h	19.10.2019		
ETHME - Status Update IV	1,50 h	21.10.2019	week IV	23,00 h
Figuring out Wallet bug & design new solution -> look at Status IV	5,75 h	22.10.2019		
Working on new CipherUtils class & refactoring Wallet class	3,50 h	23.10.2019		
Implementing a lot on CipherUtils & Wallet class, finishing lot of functionality Able to store WalletData fully encrypted and to decrypt/access it	13,25 h	24.10.2019		
Redesign and some UI functionalities at wallet creation activities	3,75 h	26.10.2019		
Finally finished up whole Wallet creation process. Including: storage, encryption, redesign, activities & clean code - *a15ef22	8,50 h	27.10.2019		
ETHME - Final Report V	2,50 h	28.10.2019	week V	37,25 h

Where are we now?

Shortly before writing this final report, the implementation of the wallet generation process was finally completed. With all its requirements regarding, having a secure way of storing the keys in an encrypted format, using both, the systems AndroidKeystore and an user's PIN as authentication method. But also having a clear, cleanly implemented process for the User, including a notice to the most important information about the recovery seed and it's importance and a simple easy to use UI.
Since a quite critical bug was found at the last status update (IV). A lot of rethinking and reimplementing needed to be done to get to the solution we have right now.
In the backend the Wallet & CipherUtils class were also implemented in the way, that it should already support signing transactions, even though this functionality isn't implemented yet in any way as a user interface. And as so, to support further hierarchical deterministic ethereum addresses, generated from the stored seed.
In general, there is to say, that focus was placed on 'clean' and easily reusable / extendable code, in order to reuse a lot of the already implemented functionalities for the next tasks, as sending ether, authenticate the user for diverse wallet actions via the PinPad class and maybe even encrypt further user data for improved privacy reasons. Like Transaction logs in the cache.

What is missing and why?

Obviously, those results aren't covering all the features, we wanted to have for that project and defined in the Whitepaper. Since we planned to not just have a wallet and be able to send and receive some Eth with that, but also to deploy ERC20 contracts on the blockchain. And all you can currently see is not more but a simple ethereum address.

The good point is, that's already enough to be able to receive some eth or any token. Since you've got a valid standardized ethereum address, with a perfectly fine working BIP32/39 mnemonic recovery seed and a securely stored and truly random generated key in the backend. Which of course seems to be the most critical part about that project.
Without a decent security layer managing the key, I wouldn't like to have some crypto/blockchain assets being managed by that wallet.

What do I mean with a decent security layer? As far as it was possible in the setting of that project, I tried my best to learn and really understand what exactly is going on behind those used libraries, as for instance web3j, javas SecureRandom or Androids AndroidKeystore system. Also, to understand how the wallet standards from BIP32/39/44… work. So, if there didn't happen a major misunderstanding somewhere in between those different technologies, I am pretty confident, that the developed solution for the key management has a decent level of security. Assuming, that it's not so easily possible to read out the RAM cells of the App during sensitive wallet data procession.

Since we now can see the importance of a solid well considered wallet/key management it might be more recognizable that, this part probably was the biggest task for that project.

However, we're still missing the whole network part. Which is about communicating with an ethereum node, polling for incoming transactions, requesting the accounts current balances, caching this kind of data in the App and displaying it practically.
And of course also sending out TXs as normal ether transfers over to ERC20 deployments.

At the beginning of the project, a few short tests were made to communicate with the network, or better to say the Infura node. Also, some web3j libraries example have been analysed a little. This part is still not that small, but I can imagine, it wouldn't/won't take that long, like the development of the wallet system did.
Since all of the above called missing features are part of the network connections and just different kind of requests. I'd guess for the real basics as displaying the accounts ether balance & sending some ether with default transaction settings, it won't take longer than 2-3 working days to have at least a proof of concept of that functionalities. Depending on what surprises in either direction could arise.

But I'm still not that sure right now how much effort it would actually be to implement these features in a clean way with all the needed UI around it and also the difference with a standardized working ERC20 smart contract.
I can imagine at least two more weeks as we had them during the project time, which would mean something in between 50-75h more working hours until we had our MVP ready with a reasonable working UI and well written code.

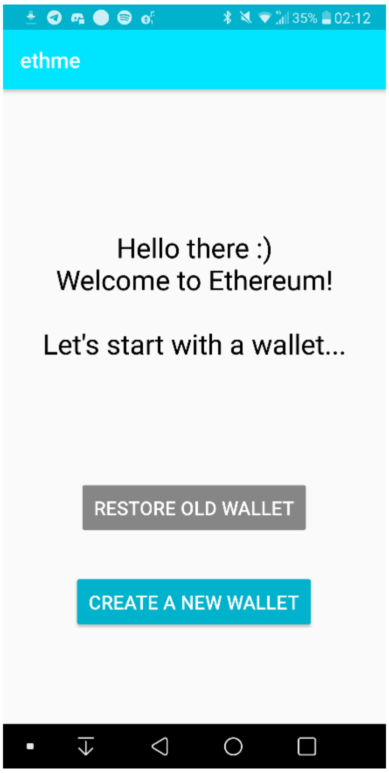
Conclusion

The project so far was a really interesting task with a steep learning curve, since a lot of new technologies were introduced to me with it. And a lot of research was done.
I also really liked to work on it and to finally work on an Android App at all, since I already wanted to do something like this for quite a while, but never found the good setting between motivation and time, which was now solved for me. As part of my study.
Especially the technology behind blockchain and cryptocurrencies is really interesting for me, and I'm glad to finally dive into it on an even deeper level as developer.

Regarding the result, I wouldn't say it was failed in anyway, but simply underestimated in the matter of complexity and therefore time effort, it needs.

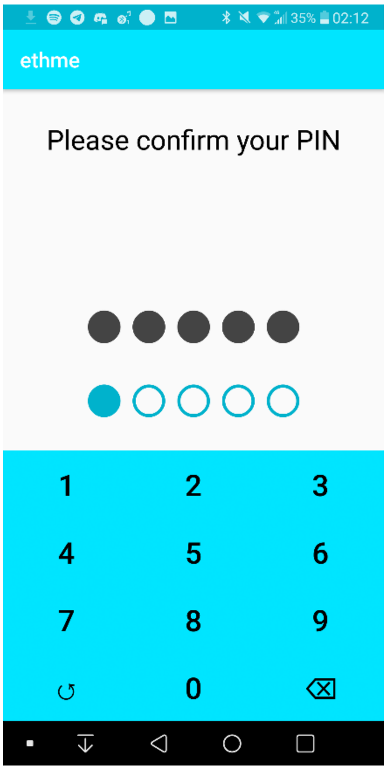
I am also happy to have a nice little open source project with that, which I'm also willing to continue a little bit more in my spare time, as far as I can find the time besides my bachelor's degree of course.

Screenshots of the whole Apps activities / states at current development:

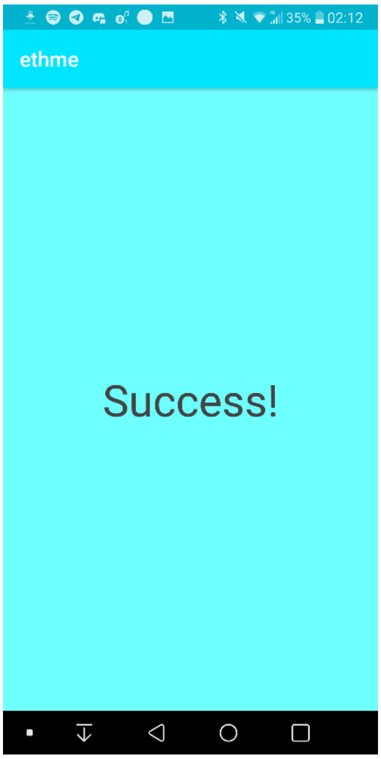


Hello there :)
Welcome to Ethereum!

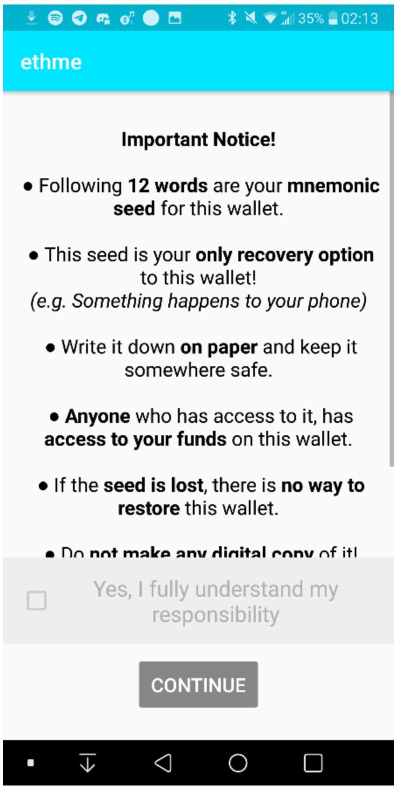
Let's start with a wallet...



Please confirm your PIN



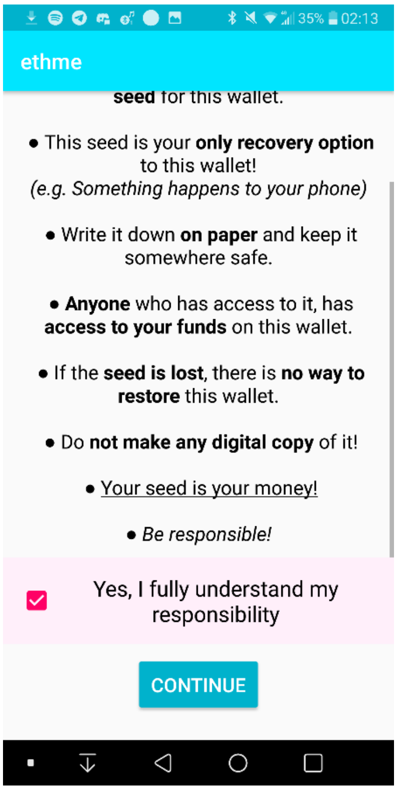
Success!



Important Notice!

- Following **12 words** are your **mnemonic seed** for this wallet.
- This seed is your **only recovery option** to this wallet! *(e.g. Something happens to your phone)*
- Write it down **on paper** and keep it somewhere safe.
- **Anyone** who has access to it, has **access to your funds** on this wallet.
- If the **seed is lost**, there is **no way to restore** this wallet.
- Do **not make any digital copy** of it!

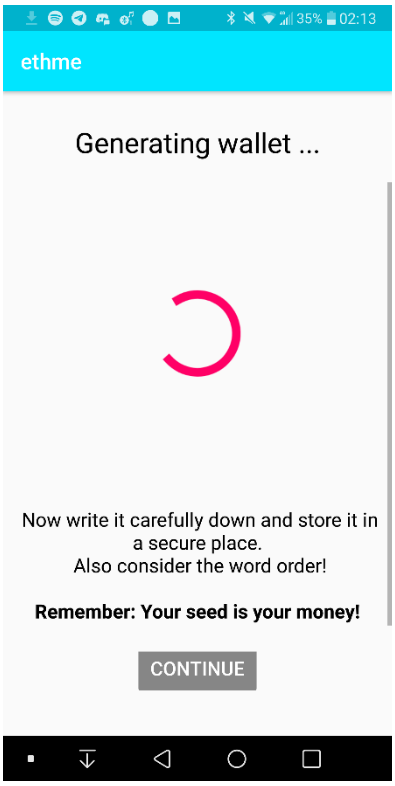
□ Yes, I fully understand my responsibility



seed for this wallet.

- This seed is your **only recovery option** to this wallet! *(e.g. Something happens to your phone)*
- Write it down **on paper** and keep it somewhere safe.
- **Anyone** who has access to it, has **access to your funds** on this wallet.
- If the **seed is lost**, there is **no way to restore** this wallet.
- Do **not make any digital copy** of it!
- Your seed is your money!
- *Be responsible!*

☑ Yes, I fully understand my responsibility

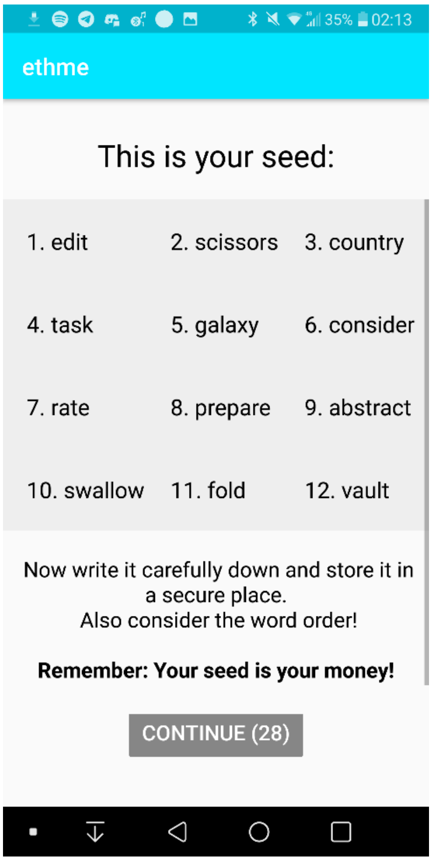


Generating wallet ...

Now write it carefully down and store it in a secure place.
Also consider the word order!

Remember: Your seed is your money!

**Any seeds or addresses seen in the screenshots shouldn't be used anyway, since they're publicly available!*

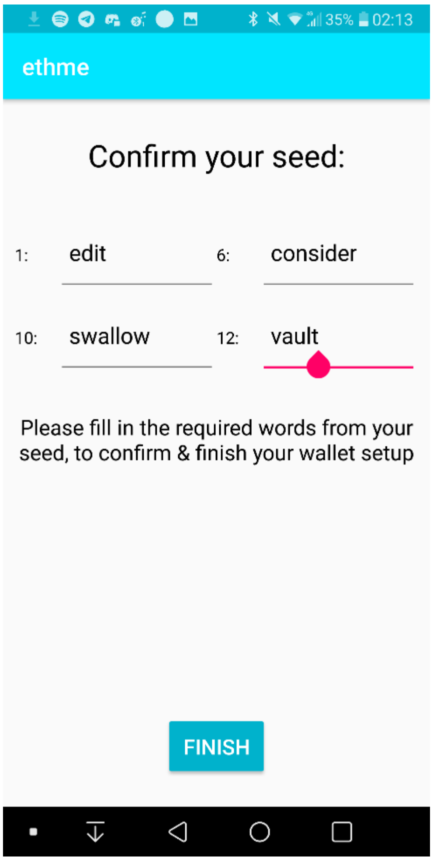


This is your seed:

1. edit	2. scissors	3. country
4. task	5. galaxy	6. consider
7. rate	8. prepare	9. abstract
10. swallow	11. fold	12. vault

Now write it carefully down and store it in a secure place.
Also consider the word order!

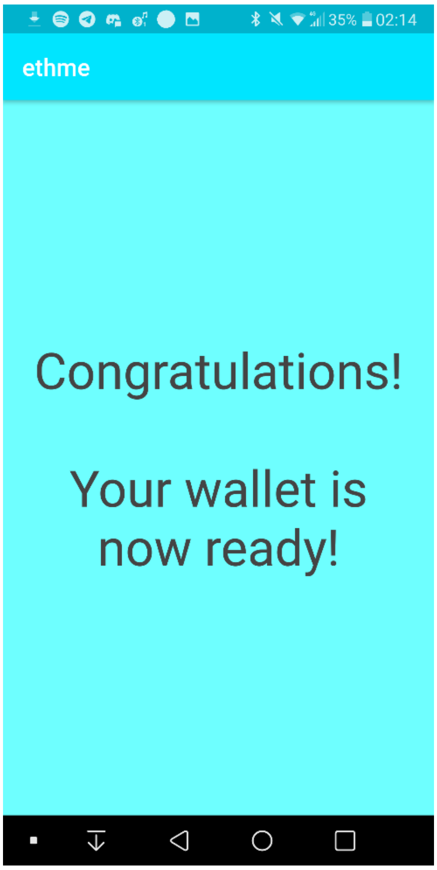
Remember: Your seed is your money!



Confirm your seed:

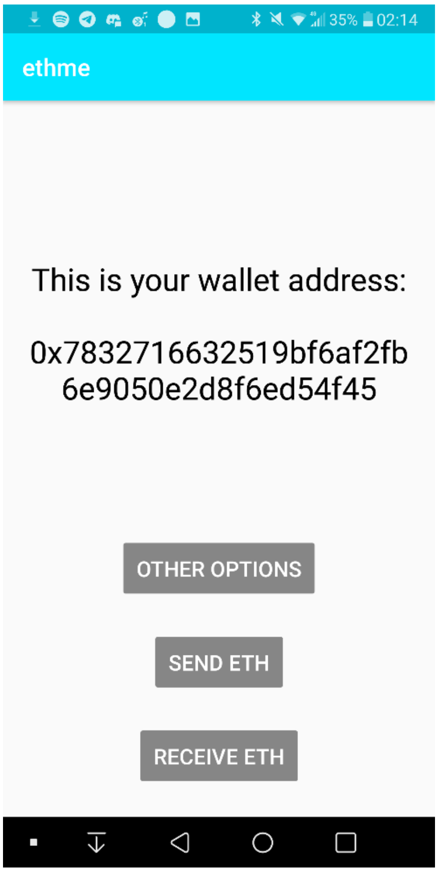
1: edit
6: consider
10: swallow
12: vault

Please fill in the required words from your seed, to confirm & finish your wallet setup



Congratulations!

Your wallet is now ready!



This is your wallet address:

0x7832716632519bf6af2fb6e9050e2d8f6ed54f45